# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*



**1. REPORT DATE** *(DD-MM-YYYY)*
22-02-2011

**2. REPORT TYPE**
Final Report

**3. DATES COVERED** *(From - To)*
01-Dec-2009 to 30-Nov-2010

**4. TITLE AND SUBTITLE**
Sensor Network Optimization by Using Error-Correcting Codes

**5a. CONTRACT NUMBER**

**5b. GRANT NUMBER**
FA9550-09-1-0289

**5c. PROGRAM ELEMENT NUMBER**

**6. AUTHOR(S)**
Xinmiao Zhang

**5d. PROJECT NUMBER**

**5e. TASK NUMBER**

**5f. WORK UNIT NUMBER**

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

Case Western Reserve University
10900 Euclid Ave.
Cleveland, OH 44106-1712

**8. PERFORMING ORGANIZATION REPORT NUMBER**

**9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
Air Force Office of Scientific Research
875 N. Randolph St. Room 3112
Arlington, VA 22203

**10. SPONSOR/MONITOR'S ACRONYM(S)**
AFOSR

**11. SPONSOR/MONITOR'S REPORT NUMBER(S)**
AFRL-OSR-VA-TR-2012-0530

**12. DISTRIBUTION / AVAILABILITY STATEMENT**
Approved for public release; distribution is unlimited.

**13. SUPPLEMENTARY NOTES**

**14. ABSTRACT**
The objective of this project is to develop efficient very large scale integrated (VLSI) error-correcting encoders and decoders to be used in sensor network applications. In addition, decoders with adjustable error-correcting capability will be designed so that trade-offs can be made on signal transmission power and receiver power. During this project period, novel schemes were proposed to further increase the speed and reduce the area of the interpolation-based generalized minimum distance (GMD) decoder of Reed-Solomon (RS) codes. In addition, the complexity analysis of the GMD decoder was generalized to take into account different codeword length, code rate and parallel processing factor. A reduced-complexity parallel interpolator has also been developed for the low-complexity Chase (LCC) algebraic soft-decision RS decoding algorithm. Compared to previous designs, it not only has much smaller area, but can run at higher speed. Adopting this design, the overall decoder can achieve substantially higher efficiency. Another achievement of this year is that the LCC decoder has been optimized, so that it can also carry out high-speed hard-decision decoding with negligible hardware overhead.

**15. SUBJECT TERMS**
Error-correcting codes, Interpolation, Reed-Solomon codes, VLSI design

| **16. SECURITY CLASSIFICATION OF:** | | | **17. LIMITATION OF ABSTRACT** | **18. NUMBER OF PAGES** | **19a. NAME OF RESPONSIBLE PERSON** |
|---|---|---|---|---|---|
| **a. REPORT** | **b. ABSTRACT** | **c. THIS PAGE** | | | Xinmiao Zhang |
| U | U | U | SAR | 7 | **19b. TELEPHONE NUMBER** *(include area code)* (216)368-6133 |

# Final Report

# Sensor Network Optimization by Using Error-Correcting Codes

## AFOSR Award Number: FA9550-09-1-0289

Time period covered by this report: 12/01/2009-11/30/2010

PI: Xinmiao Zhang
Dept. of EECS
Case Western Reserve University
Cleveland, OH 44106
E-mail: xinmiao.zhang@case.edu
Phone: (216)368-6133

Submitted to Dr. Jon Sjogren
Air Force Office of Scientific Research (AFOSR)
Physics and Electronics
875 North Randolph Street
Suite 325 Room 3112
Arlington, VA 22203

February 22 2011

## 1. Objectives

Wireless sensor networks are playing more and more important roles in many military and civilian applications. The lifetime and performance of sensor networks are dependent to a large extent on the limited power resource of individual nodes. Signal transmission requires power that is at least proportional to the square of the distance between the communicating nodes. Although multi-hop schemes can be used to reduce the distance of each communication, signal transmission still accounts for a significant part of the power consumption in sensor nodes, especially those far away from all other nodes. Error-correcting codes, a powerful tool for transmission power reduction in traditional communication systems, have been ignored in sensor networks in the past due to the perception of high complexity and negative impacts on energy efficiency. Although error-correcting encoder and decoder incur extra power consumption, they can reduce the signal transmission power substantially without compromising the data accuracy. As a result, employing carefully designed error-correcting encoders and decoders can significantly improve the overall energy efficiency.

The goal of this project is to develop efficient very large scale integrated (VLSI) error-correcting encoders and decoders to be used in sensor network applications. In particular, the focus will be given to Reed-Solomon (RS) codes, which have simple encoders and superior burst-error correcting capability. More importantly, algebraic soft-decision decoding (ASD) algorithms have been developed recently for these codes. Besides achieving better performance-complexity tradeoff than previous decoding approaches, these algorithms also have the advantage that the error-correcting capability can be easily tuned according to channel condition, as well as speed and power consumption requirement of the application. Nevertheless, mapping these algorithms directly to hardware implementation would result in high complexity. One key innovation in this project is to adopt integrated algorithmic and architectural optimizations to make the en/decoders achieve higher speed, smaller area, and/or lower power consumption in an unprecedented manner.

During the first project period (Apr. 2009-Nov. 2009), efficient implementation architectures were developed for the interpolation-based generalized minimum distance (GMD) decoder, which incorporates reliability information from the channel into erasure decision. In addition, an interpolation-based hard-decision decoder was proposed to change the perception that it has much higher complexity than traditional syndrome-based hard-decision decoders. We also analyzed how the complexities of several major ASD decoders change with various parameters, such as codeword length, code rate, maximum interpolation multiplicity, test vector number and channel condition. The decoders analyzed include the Koetter-Vardy (KV), the bit-level generalized minimum distance (BGMD), and the low-complexity Chase (LCC) decoders.

## 2. Accomplishments

This year, novel schemes were proposed to further increase the speed and reduce the area of the interpolation-based GMD decoder. In addition, the complexity analysis of the GMD decoder was generalized to take into account different codeword length, code rate and parallel processing factor. A reduced-complexity parallel interpolator has also been developed for the LCC ASD algorithm. Compared to previous designs, it not only has much smaller area, but can run at higher speed. Adopting this design, the overall decoder can achieve substantially higher efficiency. Another achievement of this year is that the LCC decoder has been optimized, so that it can also carry out high-speed hard-decision decoding with negligible hardware overhead.

**a. High-speed interpolation-based GMD decoder**

For an (n, k) RS code, k-symbol message blocks are encoded into n-symbol codewords. Traditional hard-decision decoders of RS codes can correct up to t=(n-k)/2 errors. On the other hand, the received symbols with low reliability can be set as erasures. Error-and-erasure decoders of RS codes can correct any combination of $\sigma$ errors and $\rho$ erasures, provided that $2\sigma+\rho\leq n-k$. The GMD decoding carries out error-and-erasure decoding for t+1 erasure patterns. In the ith ($0\leq i\leq t$) pattern, the 2i least reliable symbols are erased. As a result, it can achieve significant coding gain over conventional hard-decision decoding.

In the first project period, an interpolation-based one-pass decoder was developed for implementing the GMD algorithm. After applying the re-encoding and coordinate transformation, which are complexity-reducing techniques, the error-erasure locator and evaluator for the erasure-only case can be directly derived. Then the error-erasure locator and evaluator for each additional erasure pattern can be computed after two iterations of the interpolation. Compared to existing schemes that are based on the Berlekamp-Massey algorithm (BMA) and start with the error-only case, our GMD decoder can achieve significantly higher efficiency because the polynomials involved in our approach has much lower degree. Sending all of the t+1 error-erasure locators and evaluators to the rest decoding steps leads to very high hardware complexity. Hence, a polynomial selection scheme needs to be employed to pick the correct locator and evaluator. Our proposed scheme was to select the locator whose degree equals its root number. Although it is much simpler than prior approaches, it still requires exhaustive Chien search. Moreover, the Chien search needs to be finished before the next locator is computed by the interpolation step. To avoid using hardware-demanding highly parallel Chien search, the worst-case latency is considered for each iteration of the interpolation in our previous design. As a result, the interpolation is not allowed to run at full speed, and it also limits the maximum achievable throughput of the overall decoder.

By making use of the properties of the interpolation algorithm, a novel scheme was developed in this project period to enable the interpolation run at full speed without incurring large hardware overhead in the polynomial selection. In the interpolation-based GMD decoder for high-rate codes, there are two bivariate polynomials involved in the interpolation step. The latency of an interpolation iteration is decided by the maximum x-degree of the bivariate polynomials in that iteration. The maximum x-degree starts with zero, and can be increased by at most one in each interpolation iteration. After every two iterations, the bivariate polynomial of lower weighted degree consists of the error-erasure locator and evaluator. Therefore, in order to keep up with the speed of the interpolation, higher level parallel processing needs to be adopted in the Chien search for polynomial selection over the locators generated earlier. Fortunately, the degrees of the earlier locators are also lower. Based on these observations, an efficient Chien search architecture with variable parallel processing factor was developed. The proposed architecture connects those less significant polynomial coefficients to more constant multipliers, so that the Chien search for a lower-degree polynomial can be done faster without requiring extra multipliers for those more significant polynomial coefficients. Compared to the previous approach that adopts a fixed parallel processing factor, the proposed architecture requires much less area to achieve the same speed. To further reduce the area, substructure sharing has been exploited over the multipliers that share the same inputs. In addition, when the degree of the locator is one, its root number must be one. Hence, the corresponding Chien search can be skipped to reduce the power consumption.

Optimizations have also been carried out for other blocks of the interpolation-based GMD decoder. The roots found from the polynomial selection are the error locations for the k most reliable code positions. Previously, they are stored and used directly in the following steps. However, multiple roots can be found in each clock cycle due to the parallel Chien search adopted in the polynomial selection. Storing all possible roots requires large memory. Alternatively, we propose to only count the number of roots, but not store the roots, during the polynomial selection. Although a serial Chien search is needed afterwards to recalculate the roots of the chosen error-erasure locator, it requires much smaller area than the memory needed for storing all roots computed in parallel during the polynomial selection. After the errors in the k most reliable code positions are corrected, another erasure decoding is necessary to recover the rest n-k codeword symbols. The Chien search can be carried out over finite field elements in reverse order by changing the constant inputs of the multipliers. As a result, it can be pipelined with the syndrome computation in the erasure decoding. Hence, the buffer between these two steps can be eliminated. Moreover, the erasure decoding has more clock cycles to spend, and its area can be reduced accordingly. Employing these modifications, as well as the Chien search architecture with variable parallel processing factor for polynomial selection, the interpolation-based GMD decoder can achieve 50% higher speed than previous designs with negligible area overhead for a (255, 239) RS code.

The hardware complexity analysis of the interpolation-based GMD decoder was generalized. The complexity of each decoder block is expressed in terms of codeword length, code rate, and corresponding parallel processing factors. With the help of our analysis results, the interpolation-based GMD decoder can be easily adopted by various systems that use different RS codes and require different speed-area tradeoffs. In addition, the generalized analysis provides insights on how the decoder complexity changes with code parameters.

**b. Reduced-complexity parallel interpolator for the LCC ASD decoding**

In the LCC algorithm, decoding trails are carried out on $2^e$ test vectors consisting of points of multiplicity one. Here e is a positive integer. The interpolation needs to be done for each test vector. However, starting the interpolation from the beginning for each vector leads to overwhelming hardware complexity, especially when e is large. Alternatively, the test vectors can be arranged in an order so that adjacent vectors are only different in one point. Given the interpolation result of one vector, that of the next vector can be derived by employing one iteration of the backward interpolation to delete a point and one iteration of the forward interpolation to add the point that is different. To achieve additional speedup, the computations in the backward and forward interpolations can be carried out together in a look-ahead manner using a unified architecture. Despite all these efforts, the latency of the LCC interpolation is still exponential to e. On the other hand, larger e leads to better error-correcting performance. To further reduce the interpolation latency, the test vectors can be divided into groups, in each of which the vectors can be still ordered so that there is one different point in adjacent vectors. One unified interpolator can be used for each group to carry out the interpolation in parallel. However, employing multiple unified interpolators results in significant increase in the decoder area.

A novel scheme was proposed to achieve parallel interpolation with reduced area requirement. The test vectors are still divided into groups. However, the interpolation is carried out over the points in a different order. First the points that are common to the vectors in different groups are interpolated over. This can be done by a single unified backward-forward interpolator. Then the remaining points in each vector are added to the interpolation result of the common points using forward interpolators. Although multiple forward interpolators are needed, each of them only accounts for half the area of a unified interpolator. In addition, these forward interpolators can be simplified. Each iteration of the interpolation consists of polynomial evaluation and polynomial updating. The points to be added by the forward interpolators are known at the beginning of the decoding. Hence, instead of carrying out the expensive polynomial evaluation, the evaluation values can be derived through updating initial values, which can be easily determined from the initial polynomials for the interpolation. Moreover, although two polynomials are involved in the interpolation of the LCC decoding for high-rate codes, only the one with lower weighted degree will be picked as the interpolation output. Therefore, only the polynomial with lower weighted degree needs to be updated in the last iteration of the forward interpolation for each test vector. Another advantage of the proposed interpolation scheme is that the forward interpolators for adding successive points in a vector can be pipelined. Hence, the updated polynomials from one forward interpolator can be sent to the next forward interpolator right away without being stored. Accordingly, the memory requirement can be reduced. Compared to using four unified interpolators in parallel in the LCC decoding for a RS (255, 239) code with 32 test vectors, using the proposed scheme can achieve higher speed and 30% area reduction. The proposed interpolation architecture was adopted to develop LCC decoders and further optimizations were carried out. For the same RS code, the proposed LCC decoder can achieve 31% higher efficiency in terms of speed-over-area ratio compared to the decoder that employs four unified interpolators.

By changing e, the error-correcting capability of the LCC decoder can be adjusted. LCC decoders with variable e can counter the effect of time-varying channel. The proposed interpolation architecture can be also used to reduce the area of such adaptive LCC decoders. Assume that the maximum and minimum e used are e_max and e_min, respectively. The test vectors can be divided into groups of $2^{e\_min}$ vectors. Still one unified interpolator is employed to take care of the interpolation over the common points of the test vectors from different groups. Then the interpolation over the rest e_max-e_min points in each vector are completed by forward interpolators. In the case that e<e_max is used, unnecessary forward interpolators can be shut down to save power. Nevertheless, the interpolation latency does not change with e in this scheme. The fixed latency greatly facilitates making decisions on the parallel processing factors to be used in other steps of the decoding.

**c. Integrated high-speed hard-decision/ASD decoder**

Although hard-decision decoding can not correct as many errors as ASD algorithms, it consumes much less power and has much shorter latency. Therefore, ASD decoding can be activated only after the hard-decision decoding fails in order to reduce the average power consumption and decoding latency. Instead of having separate decoders, hardware units can be shared between ASD and hard-decision decoding to reduce the area requirement. To maximize sharable units, the interpolation-based algorithm is chosen over traditional syndrome-based algorithms for hard-decision decoding. Despite that the LCC decoding with one test vector is equivalent to hard-decision decoding, further modifications can be done on the LCC decoder to reduce the latency when it is used for hard-decision decoding.

As it was mentioned previously, the re-encoding technique can be adopted to reduce the complexity of interpolation-based decoders. The basic idea of re-encoding is to find a codeword that equals the received word in the k most reliable code positions. Hence, this technique was actually implemented as erasure decoding in the LCC decoder. In the case of hard-decision decoding, each code position is treated with equal reliability. Hence the first k code positions can be picked for re-encoding. In this case, the re-encoding is reduced to systematic encoding, which can be implemented by a linear feedback shifter register architecture. In the re-encoder of the

LCC decoder, a polynomial multiplication is carried out to compute the erasure evaluator polynomial. Such computation is done by an architecture similar to that of an finite impulse response filter. It also consists of multiplier-adder pairs. Therefore, multiplexors can be added to this architecture to implement systematic encoding. After the interpolation, the LCC decoder uses the Chien search and Forney's algorithm to correct the errors in the k most reliable code positions. Then another erasure decoding is required at the end to recover the entire codeword. These steps can be also simplified when the LCC decoder is used for hard-decision decoding. Using systematic encoding, the first k symbols in the codeword are the message symbols. Therefore, the Chien search only needs to be done for the first k code positions. After the errors in these positions are corrected, the messages can be recovered. Hence the erasure decoding at the end is no longer needed. Employing these optimizations, the hard-decision decoding can be completed 35% faster for a (255, 239) RS code compared to using the LCC decoder directly for hard-decision decoding. In addition, the area overhead for implementing the proposed modifications is negligible.

## 3. Publications

[1] J. Zhu and X. Zhang, "Efficient generalized minimum-distance decoders of Reed-Solomon codes" submitted to *Springer Journal of Signal Processing Systems*.

[2] X. Zhang and J. Zhu, "Reduced-complexity multi-interpolator algebraic soft-decision Reed-Solomon decoder," *Proc. of IEEE Workshop on Signal Processing Systems*, pp. 398-403, San Francisco, Oct. 2010.

[3] J. Zhu and X. Zhang, "Efficient Reed-Solomon decoder with adaptive error-correcting capability," *19th Annual Wireless and Optical Communications Conference*, Shanghai, China, May 2010.

## 4. Personnel Supported

Xinmiao Zhang (PI), Associate Professor
Jiangli Zhu, Research Assistant
Yu Zheng, Research Assistant

## 5. Conference Presentations

X. Zhang and J. Zhu, "Reduced-complexity multi-interpolator algebraic soft-decision Reed-Solomon decoder," *IEEE Workshop on Signal Processing Systems*, San Francisco, Oct. 2010.

J. Zhu and X. Zhang, "Efficient Reed-Solomon decoder with adaptive error-correcting capability," *19th Annual Wireless and Optical Communications Conference*, Shanghai, China, May 2010.

## 6. Patent Disclosures

None

## 7. Recognitions

The PI received the Research Award of the EECS Department at Case Western Reserve University last year. In addition, our work on RS decoder design is well-recognized. The PI is the only hardware designer that was invited to give a talk in the Information Theory and Applications Workshop for four consecutive years. This prestigious workshop is attended by top information and communication theorists.